

DESIGN KNOWLEDGE CAPTURE FOR THE SPACE STATION

K. R. Crouse
NASA/Johnson Space Center
Mail Code EF5, Houston, Texas 77058

D. B. Wechsler
The MITRE Corporation
1120 NASA Road One
Houston, Texas 77058

INTRODUCTION

NASA's Space Station Program (SSP) is approaching its System Design Review milestone. NASA's SSP development community is preparing to begin detailed design; to be followed by testing, evaluations, and launch. The first elements of the Space Station are planned for launch in 1994. The development environment for generation and capture of design information over this extended period will be characterized by:

- a) Continuing engineering efforts resulting in many changes.
- b) Personnel phasing in and out of the program.
- c) Technology developments bringing opportunities to apply design information in powerful and hardly-imagined ways.

The challenge of design knowledge capture will be to create and populate a base of design knowledge, to provide a sufficient foundation of collective technical memory for support of applications of the year 2000 and beyond. NASA and The MITRE Corporation have identified a technical approach and implementation plan for the capture and storage of design knowledge.

NEEDS AND BENEFITS

The capture, storage, and availability of design knowledge can benefit the Space Station Program throughout its life cycle.

Continuing Engineering and Future Designs

The importance of integration through a common database is emphasized in "team engineering" approaches to reducing product design lead time. In a team engineering environment, the work of an individual will affect many others on a project. As the size of the team

increases, cooperative interaction leads to productivity improvement, while standalone solutions reach a point of diminishing returns, due to the overhead costs of handling redundant information. As the team development of the Space Station progresses, increasingly complex engineering models will be exercised with "what-if" simulations. If root design values and inputs are retained, together with a definition of the engineering analysis descriptions, then the same or similar analysis types can be reproduced using different parameters. If the same analysis program is available, the original analysis can be re-created.

Captured and retained design knowledge can also provide a basis for solving new design problems, by using design rationale to replay design histories of similar problems. If a goal in the old problem was met using some plan, and the reasons the old design worked also hold true in the new problem; then the plan can assist in determining the new solutions as well.

Manufacturing

The integration benefits of design knowledge capture are embodied in the term "CIM" (Computer-Integrated Manufacturing), adopted in principle by industrial corporations worldwide.

The CIM concept involves intelligent combination and use of "... computer and information/communication technologies to effectively integrate all of:

- a) the engineering/design functions,
- b) the manufacturing planning functions,
- c) the equipment/process technologies,
- d) the manufacturing control processes, and
- e) the management functions

necessary to convert raw materials, labor, energy, and information into a high quality, profitable product, within a reasonable amount of time." [16]

The Corporate CIM Committee at Garrett Corporation has observed that the manufacturing organizations operating in 1995 and beyond will be fundamentally different from today's typical manufacturing companies. CIM technologies will make well-defined long-range CIM plans essential to the maintenance of competitive position.

Logistics and Field Operations

In its introduction to the Integrated Design Support Project, the U. S. Air Force Logistics Command reports: [14]

"In its wake, high technology has created massive amounts of technical information-- a mountain of paper which today must be managed manually... engineering technical data is volatile, complex, iterative, and addressable by a variety of applications. These special requirements make manual data-handling extremely labor- and cost-intensive. To date, application has been focused on design and manufacturing, with no consideration being given to integrating the overall... life cycle process."

On-Board Applications

The initial findings of NASA's Automation and Robotics Panel stated that: [3]

"... Shuttle operators rely heavily on paper backup for every mission. This mass of documents must be condensed, coordinated, and unified into a usable database if the Space Station is to reach its planned level of capability."

When the actual behavior of a system fails to match its intended behavior, the reason for failure is more easily localized if a record is available of how the system specification was decomposed and implemented. However, this debugging cannot be conducted using the design definition only. The designer's knowledge must also be applied.

CURRENT STATE

Capture is the process of obtaining information for retention in computer-interpretable form. Captured information is organized in meaningful context for

retention and use. The definition of capture does not include information held in manual media. Neither does it include data entered into storage without association of meaning, such as a scanned code, or unidentified text string. Additional conversion or interpretation of these data forms is required.

With increased use of CAD/CAM, the electronic retention of design definitions is on the increase. But near-term emphasis of Computer-Aided Design (CAD) as a drafting tool can divert attention from the need to effectively organize design definitions. Chester Fleszar, an Applicon Marketing Product Specialist, elaborates: [11]

"The problem is that we act as if we're making blueprints rather than parts. Once a company becomes involved in CAD/CAM, the electronic representation of the part becomes all-important, while the blueprint becomes obsolete. Translating all the information on the original paper drawing does nothing to improve manufacturing quality or efficiency..."

Manufacturing industry has recognized the benefit of integrating conventional CAD/CAM applications through design definition information in engineering and product databases. But for planning and logistics organizations, field operators, and customers, the design definition alone is inadequate. Users of this information must adjust to the information shortfall by attempting ad hoc to manipulate the available information, or by expending extra resources to collect the needed information.

Nevertheless, efforts to capture the designer's knowledge are rare. For the NASA community, unless the knowledge of SSP designers and engineers is captured, the SSP collective technical memory expected to be available will diminish with the development team's decreasing accessibility over time.

DESIGN KNOWLEDGE REPRESENTATION

Overview

Design knowledge encompasses not only what designs are, but how and why they satisfy the design's functional requirement. Design knowledge is represented as a linked design object structure. This "object-organized" structure is composed of design objects, object attributes, and declarations or

assertions called "designer's knowledge".

Design Object Structures

The principal unit of design knowledge organization is the design object. The linkage of design objects defines the design arrangement. Design objects are defined over a range of abstraction levels, in which each object is linked with its constituents. For example, a representation for a physical design object of a component assembly is decomposed to the constituent components. The components are decomposed to constituent features of each component. The features can be implemented as graphical elements, if the feature has a visual interpretation.

The definitions and values of each object's attributes are contained within that object's structure.

Graphics of the Design Object

The U. S. CAD/CAM community has in cooperation defined a computing system-independent means of exchanging CAD graphics files. This evolutionary standards development, coordinated by the National Bureau of Standards, is called Initial Graphics Exchange Specification (IGES). [13]

IGES defines standard data structures called entities, for geometric and other graphics-based elements. A graphics system that supports IGES can translate an IGES data structure into a geometric pictorial.

The IGES data structures corresponding with the design object's geometric elements are represented as attributes of the object. This approach allows a design object possessing this information to "draw itself", using methods including IGES translation by the CAD delivery system.

Designer's Knowledge

Design objects are the parent representations for designer's knowledge. Designer's knowledge attaches to its parent object at the highest applicable level of abstraction.

Designers' knowledge includes the information the designers used; the analysis they conducted; and the decisions they made to develop the design object. Designers' knowledge defines what the design does, and why it does so. Such definition encompasses functional and behavioral descriptions.

Examples of designer's knowledge include declarations of functional requirements

for the design object; criteria or intent for selection of a particular design approach or solution over its alternatives; declarations of analysis results and conclusions; and assertions of expected behavior in normal, marginal or failed modes.

BOUNDING OF DESIGN KNOWLEDGE

The potential size of a comprehensive SSP design knowledge aggregation suggests that approaches must be defined to avoid the capture of extraneous knowledge. The three following approaches to knowledge organization and bounding have been identified.

Bounding of Knowledge Content

The "perspective" parameter is a classification based on the use of design knowledge. Users' perspectives are defined in terms of application problems. However, since designers may both produce and use design knowledge, their analytic disciplines are also users' perspectives. Thus, a perspective could be based on either an intermediate configuration, such as a model for design analysis; or on a "flying" design, as in an on-board SSP application.

Definitions of perspective provide for the subsequent retrieval of design knowledge, by requiring that only knowledge be captured for which a perspective can be identified. Since complex future applications might involve several disciplines, perspectives can also serve to clarify the boundaries of expert knowledge in multi-discipline problem-solving.

Bounding of Knowledge Volume

The "visibility" parameter of a design object is an indicator for determining the depth of knowledge detail to be captured, and for selecting the appropriate capture tool.

Visibility is a combined valuation of the probability of failure within the design object, factored with the results of such failure. Valuations of visibility can be taken from reliability and redundancy projections. For example, if a design object (including its designed redundancies) has a high reliability and negligible results from failure, then this object will have a low visibility rating.

Bounding of Capture Frequency

The "version" parameter enables temporal support to be established in the design knowledge structure. This parameter can

be used to manage multiple knowledge versions of the same design object. Knowledge which affects a temporal value will require an accurate accounting of changes and their rationale.

THE DESIGN KNOWLEDGE SYSTEM

In the approach identified, captured knowledge will be stored in relational databases. Since applications based on advanced technologies such as object oriented programming and database inferencing are not yet in wide use, the interim step of retaining design knowledge in a relational database "object-organized" form will assure the availability of SSP design knowledge for these future application technologies.

The facilities of the database manager may be used as a capture tool. Descriptions of additional capture tools follow.

CAD and Engineering Analysis

The integration of Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE) provides a common view of the design, to facilitate reasoning and analysis about the design object. CAD provides for automation of the graphical representation process. CAE provides for automation of those engineering analysis methods used to predict the behavior of the design object. The CAE analysis perspective may be defined as a knowledge bounding parameter.

CAD/CAE Documentation

Basic documentation can be generated from within the CAD system. Attribute values may be entered when the pictorials of the design objects are created. Then the resultant information is extracted and loaded into the relational database system.

With integrated documentation, the results of the CAE analysis can be included with the CAD information. The analysis program is then easily identified, and the program itself can be referenced for future use.

Specification Language System

A Specification Language System (SLS) is a specification environment based on a formal notation for expressing design requirements in terms of function, structure, or behavioral description. The methodology of the SLS provides an organized approach for capturing design knowledge early in the development cycle, when insufficient system design definition exists to apply CAD approaches. The automated tool set of an

SLS can provide valuable assistance in assuring consistency of identifiers and terms, and in enforcing documentation and project standards.

Where well-defined relationships exist between the functional and physical definitions, certain SLS's synthesize and simulate physical structure from a functional definition. An example in VLSI circuitry is the VHSIC Hardware Description Language (VHDL). [1] Executable hardware description languages should be considered only for well-structured design problems with unambiguous physical implications.

Designer's Apprentice System

The designer's apprentice can capture design knowledge as a by-product of its interaction with the designer. A designer's apprentice may perform the following functions:

- o Suggestion of goals and constraints
- o Recognition of past successful solutions
- o Conducting and recording of system-designer dialogues
- o Assistance with tedious details

While the notion of a designer's apprentice holds promise; presently the necessary organization of design knowledge is barely understood well enough to effectively apply this tool. The strongest candidate areas are narrow domains of expertise having codified design rationale.

TECHNOLOGY FOR FUTURE KNOWLEDGE ACCESS

Accommodating Knowledge Access Technologies

Applications based on the following technologies are not in wide use. For an interim period, captured design knowledge will be arranged by object, and retained in relational storage. Advanced applications, as they are developed, will be supported with the captured SSP design knowledge, made compatible through economically tolerable modification.

Object Oriented Environment

Object-oriented technology complements design knowledge organization by characterizing systems in terms of a configuration. This approach centers descriptions around the objects that are pieced together, rather than centering on transformations of data about these systems. This organization is similar to

the linked design object structures in the database system.

Object-oriented programming holds promise for acceptance because of claims for improved programmer productivity and easier program maintenance. More important from a user viewpoint, the program organization allows those not initially familiar with the program to rapidly and accurately understand its content.

An object consists of data private to the object, and of a set of operations which can access that private data. A "consumer" object requests a "provider" object to perform one of its operations, by sending it a message telling it what to do. The provider object responds by choosing an appropriate method; executing the operation; and returning control to the consumer.

Object-oriented programming systems are now evolving into complete development platforms, including both language and database features. Commercial products are beginning to emerge.

Database-Inferencing Systems

Bridging is needed between expert system knowledge bases and database management systems.

A promising current approach involves schema translation. In comparing programming language commands with database operators and query commands, researchers have developed a mapping of schema between the two.

Schema translation may lead to development of database-inferencing systems which share schema. These systems would use a common database for a number of knowledge-based applications. The inferencing procedure would be integrated with the database management capabilities. Then application development would consist of developing the appropriate goal statements, and confirming that the supporting descriptions are in the database.

Another approach to schema translation is to locate the translation intelligence within the intelligent system development facility. In this approach, the knowledge-based application initiates a database query.

FUTURE TASKS AND ISSUES

Overview

The project of developing a design knowledge capture system involves substantial planning and preparation.

A serious implementation issue is the coordination of timing between capture system installation and Space Station development. Until capture is implemented, the risk of losing designer's knowledge is ongoing.

Many of the facilities for design knowledge will be provided as part of NASA's Technical Information Management System (TMIS). The TMIS will be used to support technical management functions of the overall Space Station Program, including the design, development, and operation of the orbital facility. The TMIS user community will include all NASA personnel involved with the Space Station, all primary contractor personnel, and all personnel representing the international partners. The TMIS resources will be based on commercial, "off-the-shelf" technology.

Following are major near-term tasks for implementation of design knowledge capture, related to computing facilities.

Relational database facility

A TMIS-compatible relational database facility will be employed as the development contractors' design knowledge repository. Adequate description of this facility will be provided in ample time to allow for development contractors' knowledge capture planning.

Standardization Issues

Standardization issues will be resolved, which arise from the resources to be provided. Such issues include common methods for CAD data exchange.

Following are major near-term tasks for implementation of design knowledge capture, related to knowledge organization.

Schemes for Knowledge Bounding

To support the development contractors' planning for capture resources and methods, initial valuations of visibility parameters will be provided for identified design objects. A classification of engineering analysis perspectives will be supplied. A common contractor approach for implementation of knowledge versioning will be defined.

Design Knowledge Content

The design knowledge base must be defined and organized, before it can be populated. Guidelines will be established for common semantics and input definitions. Available application developers will assist by providing requirements.

Existing NASA databases will be evaluated for compatibility with requirements for design knowledge content. Conforming portions will be integrated within a design knowledge context.

An evaluation of the planned content of future milestone deliverables will also be conducted, for suitability as design knowledge.

CONCLUSIONS

The benefits of design knowledge availability are identifiable and pervasive. The implementation of design knowledge capture and storage using current technology increases the probability for success, while providing for a degree of access compatibility with future applications. The Space Station design definition should be expanded to include design knowledge. Design knowledge should be captured. A critical timing relationship exists between the Space Station development program, and the implementation of this project.

REFERENCES

1. Ackley, Dave, et al., "VHSIC Hardware Description Language," IEEE COMPUTER, February, 1985.
2. Addleman, David K., Davis, Margaret J., and Presson, Edward P., "Specification Technology Guidebook," CONTRACT F30602-84-C-0073, U. S. Air Force, August, 1985.
3. Advanced Technology Advisory Committee, ADVANCING AUTOMATION AND ROBOTICS TECHNOLOGY FOR THE SPACE STATION AND FOR THE U. S. ECONOMY, NASA TECHNICAL MEMORANDUM 87566, April 1, 1985.
4. Anderson, Sandra V., Bell, John B., and Smylie, Susan E., INFORMATION ASSET MANAGEMENT, VOLUME IV, The MITRE Corporation Technical Report Number 8613, June, 1986.
5. Andrews, Timothy and Harris, Craig, "Combining Language and Database Advances in an Object-Oriented Development Environment," (To Be Published) OOPSLA '87 PROCEEDINGS, October, 1987.
6. Beazley, William G., "Available Expert System Knowledge in CAD/CAM Databases," CONTRACT F19628-86-C-0001, The MITRE Corporation, July, 1986.
7. Brown, D. C. and Chandrasekaran, B., "Expert Systems for a Class of Mechanical Design Activity," WORKING CONFERENCE ON KNOWLEDGE ENGINEERING IN COMPUTER-AIDED DESIGN, Elsevier Science Publishing Company, 1984.
8. Clancey, William J., "Heuristic Classification," ARTIFICIAL INTELLIGENCE JOURNAL, December, 1985.
9. Cox, Brad J., OBJECT-ORIENTED PROGRAMMING, Addison-Wesley Publishing Company, 1986.
10. Crouse, K. J. and Spitzer, J. F., "Design Knowledge Bases for the Space Station," PROCEEDINGS OF ROBEXS '86, June, 1986.
11. Fleszar, Chester, "Is IGES the Problem, or are We?," CIM TECHNOLOGY, Fall, 1986.
12. Genesereth, Michael R., "The Use of Design Descriptions in Automated Diagnosis," QUALITATIVE REASONING ABOUT PHYSICAL SYSTEMS, The MIT Press 1985.
13. IGES Organization, INITIAL GRAPHICS EXCHANGE SPECIFICATION- VERSION 3.0, National Bureau of Standards, April, 1986.
14. INTEGRATED DESIGN SUPPORT, U. S. Air Force Wright Aeronautical Laboratories, 1986.
15. Maier, David, et al., "Development of an Object-Oriented DBMS," OOPSLA '86 PROCEEDINGS, September, 1986.
16. Mize, Joe H., Seifort, Deborah J., and Settles, Stan F., "Formation and Workings of a Corporate-Wide CIM Committee at Garrett Corporation," INDUSTRIAL ENGINEERING, November, 1985.
17. Mostow, Jack, "Towards Better Models of the Design Process," AI MAGAZINE, Spring, 1985.
18. Rehak, Daniel R., Howard, H. Craig, and Sriram, Duvvuru, "Architecture of an Integrated Knowledge-Based Environment for Structural Engineering Applications," WORKING CONFERENCE ON KNOWLEDGE ENGINEERING IN COMPUTER-AIDED DESIGN, Elsevier Science Publishing Company, 1984.
19. Rosenfeld, Lawrence W. and Belzer, Avrum P., "Breaking Through the Complexity Barrier--ICAD," PROCEEDINGS OF AUTOFAC '85, November, 1985.

20. Steinke, George C. and Schussel, Martin D., "Engineering by the Book... and On-Line," MECHANICAL ENGINEERING, November, 1985.
21. Snodgrass, Richard and Ahn, Ilsoo, "Temporal Databases," IEEE COMPUTER, September, 1986.
22. Stefik, Mark and Bobrow, Daniel, "Object-Oriented Programming: Themes and Variations," AI MAGAZINE, Winter, 1986.
23. Sowa, J. F., CONCEPTUAL STRUCTURES: INFORMATION PROCESSING IN MIND AND MACHINE, Addison-Wesley Publishing Company, 1984.
24. Sullivan, Jerry S. and Rummler, David C., "Second-Generation CAE Tools Learn to Share One Database," ELECTRONIC DESIGN, July 10, 1986.
25. Walker, Robert A. and Thomas, Donald E., "A Model of Design Representation and Synthesis," PROCEEDINGS OF THE IEEE 22ND DESIGN AUTOMATION CONFERENCE, 1985.